


ARRA LQCD Computing

Technical Design & Performance

Chip Watson
Jefferson Lab Scientific Computing Group

May 16, 2012

LQCD ARRA Technical Goals

Performance Goal:

To significantly increase the computing resources available to the USQCD collaboration for "analysis"…

Original target was 16 Tflops sustained aggregate performance averaged over the 3 dominant inverter actions:

- Domain Wall Fermions (DWF)
- Staggered (asqtad = a-squared tadpole)
- Clover, particularly anisotropic clover

As a slight variation from the LQCD-ext project, all three actions are included in the benchmark definition.

The current system, which heavily exploits GPUs as accelerators, sustains an effective aggregate performance of 84 Tflops.

Quantifying Aggregate Performance

(Reminder) LQCD computing proceeds in 2 phases:

1. Configuration generation (on supercomputers)
   - ❖ Must be produced sequentially, at highest performance
   - ❖ End product: 1000+ configuration files
2. Analysis (propagator generation + observables)
   - ❖ 1000 + jobs able to run in parallel
   - ❖ Target performance: 1% of configuration generation (then at 10's of Tflops)

Analysis is the task relevant for this project. For benchmarking for the LQCD ARRA resources, we selected production lattice sizes for each of the 3 main inverters:

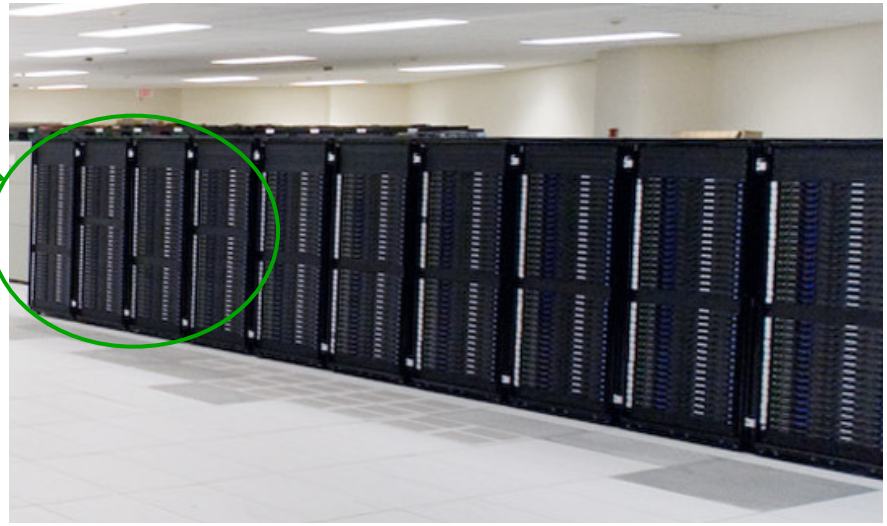
- ❖ Anisotropic Clover: 24^3 x 128
- ❖ Asqtad: 56^3 x 96
- ❖ DWF: 32^3 x 64 x 16

CPU Cluster & IB Fabric Design

The most cost effective conventional nodes were dual Intel systems,
2.4 GHz Nehalem / 2.53 GHz Westmere (phase 1 / 2), about 20 Gflops/node

QDR Infiniband switches have 36 ports, so can hold 32 nodes and still have ports free to connect to the file systems (powers of 2 are best for LQCD). Deploying multiple sets of 32 nodes reduces the cost of the Infiniband fabric while maintaining the highest efficiency for jobs up to 640 Gflops.

17 racks purchased for phases 1 & 2:
13 as single racks non-oversubscribed,
4 interconnected 2:1 oversubscribed
(to support job up to ~2 Tflops)



Most jobs on these clusters are 1 node (8 cores), 8 nodes (64 cores), or 32 nodes (1 rack, 256 cores)

Large jobs (256-1024 cores) are moved to a higher priority run queue to prevent starvation by small jobs & backfilling.

All partitions have 2 uplinks to a core switch for file services

File System

Open source Lustre was chosen to support a large flat namespace, and to enable us to scale out in capacity and performance.

Final Configuration: 416 TB, > 2 GB/s, $228K

Phase 1: 224 TB across 14 servers (excludes RAID-6 8+2 overhead)

- dual Nehalem 2.26 GHz, 12 GB memory
- 24*1TB disks, 24 disk RAID controller, DDR Infiniband
- bandwidth measured at 1.4 GB/s using 6 nodes (single DDR uplink)

Phase 2: 192 TB across 4 servers

- similar to above, but with 3 RAID-6 (8+2) strips per server instead of 2
- 2 TB disks, QDR Infiniband, higher performance RAID controller
- somewhat lower bandwidth / TB, but still more than necessary

An upgraded Meta-Data Server is now dual head with auto-failover.

GPU Accelerators

Strategy: buy as much computing capacity for the dollar as possible.

As the ARRA project was starting, USQCD collaborators were finishing up a GPU accelerated implementation of a key kernel (inverter) and were achieving high performance; therefore GPUs were incorporated into the project to increase the total performance.

Phase 1: 25% of compute funds to GPU accelerated nodes

- ✓ Enough software was becoming ready to exploit this capacity, and software development environment (CUDA) was maturing rapidly
- ✓ GPUs allowed this project to double the USQCD total computing capacity

Phase 2: 50% of compute funds to GPU nodes

- ✓ Multiple groups were in production, and were eager to absorb a large increase in capacity; allowed project to again double the USQCD total capacity
- ✓ Availability of ECC memory on the Tesla GPUs held a promise of expanding beyond inverters to satisfy more of the collaborations computing requirements; many users now exploit this capability

GPU Cluster Design

Summary of Key Decisions:

1. NVIDIA CUDA chosen as the most productive software environment.
2. NVIDIA Fermi Tesla cards are the only GPUs supporting ECC memory protection, again keeping us single supplier.
3. For some kernels, GeForce gaming cards are much more cost effective than Tesla cards, with both lower cost and higher performance (ECC on GDDR memory consumes bandwidth, plus GeForce cards are clocked higher).

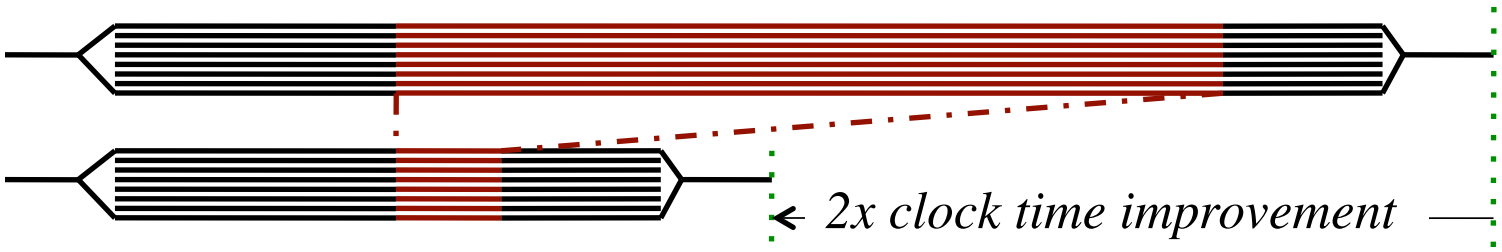
   Occasional memory errors can be caught on large matrix inversions by a quick test of the residual when the kernel has completed, so running on imperfect hardware is acceptable.
4. The early (and current) workload is mostly 1-4 GPUs with light enough use of the CPU to allow putting 4 GPUs into a single host, yielding a very high performance, modest cost platform.

Amdahl's Law (Problem)

A major challenge in exploiting GPUs is Amdahl's Law:

If only 60% of the code is GPU accelerated by 6x,

the net gain is only 2x.



Since each GPU has 6x the memory bandwidth of the then-best Xeons, 4 GPUs could reach 24x performance gain, or even higher for half-precision floating point (supported by the GPUs), giving a gain of only 2.4x on the above code.

Fortunately many LQCD codes spend > 95% of their clock time in a single kernel, a matrix inversion, and so for these applications Amdahl's Law is not a show-stopper.

Ultimate solution: we need to move more code to the GPU, and/or identify task level parallelism (overlap CPU and GPU).

Measuring GPU System Performance

The project has defined cluster performance in terms of the average of the inverters for two actions, asqtad and DWF. Unfortunately, if only the inverters are ported to the GPU, they give a poor measure of performance for accelerated nodes.

- for applications needing 80% of its x86 cluster time in the inverter, the inverter performance is a very clean benchmark
- for a quad GPU system with inverter performance 16 times higher (for example), the application only sees a clock time reduction of 4x, not the 16x for the inverter alone

To deal with this, we define an effective performance as the performance of the inverter on an x86 cluster times the job's clock time reduction, 4x in this case.

An accelerated cluster's performance, then, is dependent upon the mix of jobs it runs, and their achieved clock time reduction.

GPU Job Effective Performance (2010)

The following table shows the number of core-hours in a job needed to match one GPU-hour in a job. Last project used 32 single GPU nodes and was I/O bound.

The allocation-weighted performance of the cluster in 2010 was 63 TFlops.

Following upgrades to the cluster in late 2011, this increased to **74 TFlops**.

Project	2010-2011 Hours	#GPUs, nodes	Jpsi core hours / GPU hour (job time)	Effective Performance Gflops/node	GPU used
Spectrum	1,359,000	4, 1	180	800	(average)
thermo	503,000	4, 1	90	400	(average)
disco	459,000	4, 1	92	410	C2050
Tcolor	404,000	4, 1	40	175	GTX285
emc	311,000	4, 1	80	350	(average)
gwu	136,000	32, 32	47	50	GTX285

GPU Effective Performance

To summarize…

GPU Effective Performance is the amount of conventional Infiniband cluster that would have been needed to do the same science calculation.

$$\text{GPU E.P.} = \frac{\text{cluster\_job\_performance} * \text{cluster\_clock\_time}}{\text{GPU\_clock\_time}}$$

Different applications achieve different accelerations, and the final rating is the allocation weighted average of effective performances. In other words, the performance of the GPU depends upon the applications!

Additional Design Points

- The GPU host is ~$4K; 4 GPUs per host amortizes that cost better than 1 or 2 GPUs, but worsens the effect of Amdahl's Law (acceleration is higher)
- A survey of anticipated usage revealed a large workload that was heavily inverter dominated (95% - 99.5%); so, we mostly adopted quad GPUs
- Since Nehalem/Westmere CPUs have 36 lanes of PCI and the GPUs consume 16 lanes each, the Infiniband cards were relegated to a 4x slot, cutting their bandwidth in half
  - Most nodes were set up for single node work, with a recycled SDR card for good file I/O; this also proved adequate for dual node, 8 GPU work
  - Phase 2 Fermi Tesla nodes included QDR in 4x slot, to enable somewhat better scaling to multi-node running
- For problems with strong Amdahl's Law constraints, or needing more host memory resources, we also deployed single GPUs into one of the 17 Infiniband racks (32 nodes, full QDR bandwidth).

System Evolution 2011-2012

2011 testbed (additional 8 nodes, also running production jobs)

- dual Westmere CPUs, 48 GB memory, same as phase 2 systems
- PLX PCIe switch chips (1:2) yields 8 full PCIe2 x16 slots
- 4 GPUs attached to one CPU, to allow performance testing of GPU direct (GPU to GPU DMA without going to/from host memory)
- dual rail QDR attached to the second CPU, to enable scaling studies to 32 GPUs

Since GPU direct can't cross between the two CPUs, this system allowed stronger scaling tests of GPU direct for up to 4 GPUs.

GTX-580 testbed (now in production)

A few GTX-580 cards were procured to see how they performed compared to the GTX-480 cards, and to see how reliable they would be in comparison. The early results were encouraging; an additional 28 cards were bought to further study reliability, followed by another 160 cards to replace most of the GTX-285 cards.

GPU Comparison

Card	GPU	#cores	clock speed (GHz)	memory size (GB)	raw memory bandwidth (GB/s)	clover inverter (Gflops)[1]	cost[2]
GTX-285	GT200b	240	1.47	2	159	135	$500
GTX-480	Fermi	480	1.40	1.25	177	300	$500
GTX-580	Fermi	512	1.54	1.25	192	330	$500
C2050[2]	Fermi	448	1.15	2.67	144	185	$2100

[1] All numbers are for mixed precision

[2] Cost of C2050 has fallen since these systems were procured

[3] C2050 evaluated with ECC enabled

Notes:

C2070 and C2090 systems also exist (the latter with a full complement of 512 cores).

We will soon test a Kepler GTX-680, and eventually a Kepler Tesla card.

Fall 2011 GPU Upgrade

Because of the high performance of the GTX-580 cards, it was highly cost effective to replace the GTX-285s with GTX-580s. A change was proposed and approved. This was a level 2 change only, but still discussed with DOE and approved.

Of the total of 160 cards bought, 128 (80%) are now in production use. Some of the remaining 32 should be able to be replaced under warranty; some are ok for gaming, but not LQCD.

Even taking into account bin selecting the GTX-580s (discarding 20%), upgrading 3 nodes cost the same as buying 1 new node, and increased the performance by the equivalent of 2.5 new nodes!

This type of upgrade is only cost effective because the gaming cards are such a small percentage of the system cost (~33% after bin selecting cards).

Software Trends

Early running on the GPUs was predominantly matrix inversions, with the largest fraction of that being split half-single precision anisotropic clover. Multi-GPU running with the problem split in only one dimension (the longest, time) yields very good scaling to 8 GPUs, enough to hold the next production problem size.

The low cost of matrix inversions and the large GPU resource has moved the bottleneck elsewhere, and software developments are underway to move more of the data parallel work onto GPUs (ECC capable). These will yield performance/dollar gains lower than the initial low-hanging fruit at 10x-12x, but still worthwhile.

The second of the 3 production actions (asqtad) is now running on GPUs. Software development remains a bottleneck for growing exploitation of the GPUs.

Final Hardware Procurements

Extrapolating labor costs to the end of FY2012, the planned date for the transition of the LQCD ARRA hardware to the LQCD-ext project, there will be approximately $150K of remaining funds (3% of total project funds).

These funds will be used to augment the computational resources, with exact details of a procurement to be determined in the coming 2 months. The intent is to complement the procurements being done for the LQCD-ext project to give the greatest benefit to the USQCD collaboration.

Technical Summary

The ARRA LQCD Computing project has deployed

10 Tflops conventional infiniband systems

416 TBytes disk, backed by multi-petabyte tape library

508 GPUs equivalent to over 100 Tflops sustained capacity for anisotropic clover inverter-heavy jobs, and 74 Tflops for the mix of jobs running this year

Total deployed capacity: 84 Tflops (effective), a gain of 5x over the original plan of 16 Tflops.

The total effective Tflops depends upon the efficiency with which the applications use the GPU, and could in principle rise as a larger fraction of the existing code is ported to the GPU (reduced Amdahl's Law problem), or fall as new applications with lower GPU intensity begin to exploit the GPUs.